REMARKS

Claims 1-39 are pending in the application.

35 U.S.C. § 102 and § 103 Rejections

In the present Office Action, claims 1-7, 9-12, 13-19, 21-36, and 38-39 stand rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,195,680 (hereinafter "Goldszmidt"). Further, claims 8, 20, and 37 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Goldszmidt in view of U.S. Patent No. 6,249,801 (hereinafter Zisapel). Applicant respectfully traverses the above rejections and requests reconsideration in view of the following discussion.

In paragraph 2 of the present Office Action, the examiner suggests that Goldszmidt discloses all of the features of claim 1. In the rejection, a number of equivalences are made between features in the claim and elements disclosed in Goldszmidt. However, as discussed below Applicant submits Goldszmidt does not disclose all of the features as recited.

In the Office Action, the examiner states Goldszmidt discloses the features of claim 1 including:

> "circuitry (control server 2.1, Fig. 1a) for computing input data into a result value according to logic rule (for computing number of connection streams to each streaming server, see col. 8, lines 44-54) and for selecting a context based on the computed value (for selecting a server based on the computed number of connection streams to each streaming server, see col. 8, lines 44-54)." (Office Action, page 3).

Accordingly, the examiner equates Goldszmidt's TCP router/control server (2.1) with the recited circuitry for computing input data and selecting a context from the pool of contexts (Office Action, page 3).

The examiner further states:

> "Goldszmidt discloses a context-selection mechanism for selecting a context (selecting a streaming server based on size, capacity, location/affinity, network connectivity and utilization rate, see col. 4, lines 26-67, col. 5, lines 1-3, 55-64 and Fig. 1a)." (Office Action, page 2, para. 2).

Accordingly, the examiner equates Goldszmidt's streaming servers (1.2, 1.3) with the recited contexts which are selected by the circuitry.

Further, the examiner states Goldszmidt discloses:

> "a loading mechanism for preloading the packet information into the selected context (affinity tables are maintained in the TCP router/control server, see col. 6, lines 32-60) for subsequent processing (to maintain affinity records to indicate which node is client was routed to, see col. 6, lines 32-60)." (Office Action, page 3).

Therefore, the examiner apparently equates the disclosed affinity tables with the loading mechanism - or alternatively with the recited packet information (Office Action, page 3). However, Applicant submits neither equation is proper. The affinity tables merely indicate to which node a client (request) was previously routed. Based on the tables, a new request from a given client may then be routed to the indicated node. For example, Goldszmidt discloses:

> "The affinity-based system includes a multi-node server, wherein any of the server nodes can handle a client request, but wherein clients have affinity to one or more of the server nodes that are preferred to handle a client request. Such affinity is due to state at the servers either due to previous routing requests, or data affinity at the server. <u>At the multi-node server, a node may be designated as a TCP router. The address of the TCP router is given out to clients, and client requests are sent thereto. The TCP router selects one of the nodes in the multi-node server to process the client request and routes the request to this server</u>; in addition, <u>the TCP router maintains affinity tables, containing affinity records, indicating which node a client was routed to</u>. In processing the client request, the

> server nodes may determine that another node is better suited to handle the client request, and may reset a corresponding TCP router affinity table entry. The server nodes may also create, modify or delete affinity records in the TCP router affinity table. Subsequent requests from this client are routed to server nodes based on any affinity records, possibly combined with other information (such as load)." (Goldszmidt, col. 6, lines 40-60).

As seen from the above, Goldszmidt discloses a multi-node server wherein one of the nodes in the multi-node server is selected to serve as a "TCP router" for the remaining "server nodes" of the server. This TCP router merely receives and routes requests to one of the server nodes. The TCP router node also maintains affinity tables which may be used to make routing decisions. Additionally, the "server nodes" may modify affinity records in the "TCP router" affinity table. However, <u>the TCP router simply serves to route requests</u> and there is no disclosure of the TCP router preloading the packet information into a selected context for subsequent processing as recited.

Further, the affinity table is not equivalent to the packet information. Claim 1 recites:

> "circuitry for computing input data into a value according to one or more logic rules and for selecting a context from the pool of contexts based at least in part on the value; and
> a loading mechanism for <u>preloading the packet information into the selected context</u> for subsequent processing."

As seen in the recitation above, the packet information is preloaded <u>into the selected context</u>. In contrast, as described above, the affinity table of Goldszmidt is maintained in the TCP router and not in the selected streaming server. Goldszmidt clearly teaches that the affinity records are maintained in the designated TCP router. As noted above, the Examiner states Goldszmidt's TCP router/control server (1.1, 2.1) is equivalent to the recited circuitry, and Goldszmidt's streaming servers (1.2, 1.3) are equivalent to the recited contexts (page 2, para. 2 of the office action). Even were one to accept such equivalences, Goldszmidt clearly does not teach the affinity tables/data are preloaded from the packet into <u>the selected context</u> for subsequent processing. Rather, the

affinity tables are clearly maintained in the TCP router server (i.e., the entity which selects the server for processing) and not the selected server itself.

For at least the above reasons, claim 1 is patentably distinguishable from the cited art. Claims 13 and 26 are distinguishable for similar reasons.

With regard to Applicant's previous reply, on page 18 of the present Office Action, the Examiner states:

> "Applicant also argued that affinity tables disclosed in Goldszmidt are not maintained in a streaming server, the examiner respectfully disagrees. Goldszmidt discloses a router/server architecture that comprises streaming servers that create, modify, or delete affinity records inside the router (col. 6, lines 53-60)."

Within the entirety of Applicant's previous comments, the discussion assumed (for the sake of argument) distinctions made in both the art, and by the examiner, between the TCP router (1.1, 2.1) and the streaming servers (1.2, 1.3) which may be selected by the TCP router in Goldszmidt. Note the remainder of Applicant's statement left out of the quotation above which continues: "but rather in the TCP router/control server." It was in this sense that Applicant stated Goldszmidt does not disclose maintaining the affinity tables in a streaming server (1.2, 1.3) (i.e., one of the non-router servers). More particularly, as discussed above, <u>Goldszmidt clearly does not disclose maintaining the affinity tables in "the selected context</u>." In Goldszmidt, the TCP router maintains the affinity tables and the TCP router is not the selected context. So even if one were to characterize the router server of Goldszmidt as a streaming server, Goldszmidt still does not disclose maintaining the affinity tables in the selected context as suggested, and Goldszmidt does not disclose "a loading mechanism for preloading second input data from the data packet into the selected context for subsequent processing" as recited.

In addition to the above, the dependent claims recite additional features not disclosed or suggested by the cited art. For example, claim 6 recites "the context-selection mechanism of claim 5 wherein the first input data into the computation circuitry

further includes real time information of any processing streams stalled in un-available ones of the pool of contexts and the reason for the stall." On page 5 of the present Office Action, it is suggested that:

> "Goldszmidt discloses the context-selection mechanism of claim 5 wherein the input data into the computation circuitry further includes real-time information of any processing streams stalled in un-available ones of the pool of contexts (real-time information of the failure of a streaming server based on determining that the received bit rate, see col. 10, lines 1-18) and the reason for the stall (when the bit rate is below a threshold, see col. 10, lines 1-18)."

The cited portion of Goldszmidt is reproduced below for reference purposes.

> "FIG. 5 depicts an example of a method having features of the present invention for automatically and gracefully switching clients among multiple streaming servers in the event that a streaming server (SS) (3.2, 3.3, FIG. 3) becomes overloaded or fails. In this example, the client 3.5 is receiving a continuous multimedia stream and the switching must be transparent to the client and maintain uninterrupted playback of the multimedia streams. As depicted, in step 5.1, the client 3.5 detects a failure. By way of example only, the determination can be made based on: the received bit or frame rate (for video); a bit rate or sample rate (for audio); monitoring the delivery rate and/or for packets arriving out of order: for example using packet numbering mechanisms available in TCP; sequence numbering or time stamp capabilities of RTP (in combination with the User Datagram Protocol (UDP)). For example, RTP includes a field allowing the use of time stamps or sequence numbers. In any case, the determination could be based on the rate measurement or monitoring mechanism falling below (or exceeding) some threshold." (Goldszmidt, col. 9, line 66 – col. 10, line 18).

As may be seen from the above, Goldszmidt merely discloses that a failure or overload may be detected by monitoring a bit rate, frame rate, or packet delivery rate. When one of these rates falls below a threshold, a failure is detected. However, Goldszmidt says nothing about detecting or communicating a reason for the failure. Simply detecting a rate falls below a threshold may comprise detecting a failure, bit it does not indicate a reason for the failure. Rather, it is a condition for determining that a failure has occurred, for whatever reason. As taught by Goldszmidt:

> "If the bit rate falls below a given threshold, the connection to the server 3.2 can be considered to have failed." (Goldszmidt, col. 9, lines 12-14).

Accordingly, Applicant finds no teaching or suggestion in Goldszmidt that "first input data into the computation circuitry further includes real time information of any processing streams stalled in un-available ones of the pool of contexts and the reason for the stall," as is recited in claim 6. For at least these reasons, Applicant submits that claim 6 (and claims 18 and 35 for similar reasons) is patentably distinguishable from the cited art for at least these additional reasons.

With respect to claim 7, it is suggested that Goldszmidt discloses these features at col. 10, lines 49-63. Claim 7 recites the additional features:

> "wherein <u>the input data into the computation circuitry</u> further includes statistical data <u>about previous processing time periods required to process similar data packets</u>."

As can be seen, claim 7 recites <u>the</u> input data into <u>the computation circuitry</u>. In contrast, the cited disclosure of Goldszmidt describes enhancements to a <u>client</u> and input data into <u>the client</u>. In particular, the client may determine a server's delivery rate by comparing a server's time stamp with the delivery time of a packet. Applicant finds no disclosure herein of "<u>input data into the computation circuitry</u>" "<u>about previous processing time periods required to process similar data packets.</u>" There is no description here of previous processing time periods to process similar data packets. Further, there is no disclosure of any statistical input data into the computation circuitry. In the Office Action, the examiner's application of the disclosure to the claim merely states "delivery rate is based using server time stamps." As noted, the delivery rate is determined by the client. Should the examiner wish to maintain the rejection, Applicant respectfully requests the examiner point out clearly and with particularity how the disclosure meets all features of the claim.

In view of the above, Applicant respectfully requests withdrawal of the rejections.

CONCLUSION

Applicant submits the application is in condition for allowance, and an early notice to that effect is requested.

Respectfully submitted,

/James W. Huffman/

Reg. No. 35,549
James W. Huffman

Date: 4/25/2007